

Ponte al día con lo más destacado de Firebase en Google I/O 2023. [Más información](https://firebase.blog/posts/2023/05/whats-new-at-google-io)  
(<https://firebase.blog/posts/2023/05/whats-new-at-google-io>)

# Conecta tu app a Firebase

Si aún no lo has hecho, [agrega Firebase a tu proyecto de Android](https://firebase.google.com/docs/android/setup?hl=es-419)  
(<https://firebase.google.com/docs/android/setup?hl=es-419>).

## Crea una base de datos

1. Ve a la sección **Realtime Database** de [Firebase console](https://console.firebase.google.com/project/_/database?hl=es-419)  
([https://console.firebase.google.com/project/\\_/database?hl=es-419](https://console.firebase.google.com/project/_/database?hl=es-419)). Se te pedirá que selecciones un proyecto de Firebase existente. Sigue el flujo de trabajo para crear la base de datos.
2. Selecciona un modo de inicio para las reglas de seguridad de Firebase:

### Modo de prueba

Es el modo recomendado si recién comienzas a usar las bibliotecas cliente para dispositivos móviles y la Web, pero permite que todos lean y reemplacen tus datos. Después de que realices las pruebas, **asegúrate de revisar la sección [Comprende las reglas de seguridad de Firebase Realtime Database](https://firebase.google.com/docs/database/security?hl=es-419)**  
(<https://firebase.google.com/docs/database/security?hl=es-419>).

★ **Nota:** Si creas una base de datos en modo de prueba y no realizas cambios a las reglas predeterminadas (que se pueden leer y escribir en todo el mundo) dentro de un período de prueba, recibirás una alerta por correo electrónico y, luego, las reglas de tu base de datos rechazarán todas las solicitudes. Ten en cuenta la fecha de vencimiento durante el flujo de configuración de Firebase console.

Si quieres comenzar a usar los SDK de Apple, Android o la Web, selecciona el modo de prueba.

### Modo bloqueado

Rechaza todas las operaciones de lectura y escritura de clientes móviles y web. Tus servidores de aplicaciones autenticados aún pueden acceder a tu base de datos.

### 3. Elige una ubicación para la base de datos.

Según la ubicación de la base de datos

(<https://firebase.google.com/docs/projects/locations?hl=es-419#rtdb-locations>), la URL de la base de datos nueva tendrá uno de los siguientes formatos:

- `DATABASE_NAME.firebaseio.com` (para bases de datos en `us-central1`)
- `DATABASE_NAME.REGION.firebaseioapp` (para bases de datos en todas las demás ubicaciones)

### 4. Haz clic en **Listo**.

Cuando habilitas Realtime Database, también se habilita la API en el Administrador de API de Cloud

(<https://console.cloud.google.com/projectselector/apis/api/firebase.database.googleapis.com/overview?hl=es-419>)

## Agrega el SDK de Realtime Database a tu app

En el **archivo de Gradle** (generalmente `<project>/<app-module>/build.gradle.kts` o `<project>/<app-module>/build.gradle`) de tu módulo (a nivel de la app), agrega la dependencia de la biblioteca de Android para Realtime Database. Te recomendamos usar la BoM de Firebase para Android

(<https://firebase.google.com/docs/android/learn-more?hl=es-419#bom>) para controlar las versiones de las bibliotecas.

Kotlin+KTX

Java

Android

Android

```
dependencies {  
    // Import the BoM for the Firebase platform  
    implementation(platform("com.google.firebase:firebase-bom:32.3.1"))  
  
    // Add the dependency for the Realtime Database library  
    // When using the BoM, you don't specify versions in Firebase librar
```

```
implementation("com.google.firebase:firebase-database")
}
```

### Si usas la BoM de Firebase para Android

(<https://firebase.google.com/docs/android/learn-more?hl=es-419#bom>), tu app siempre utilizará versiones compatibles de las bibliotecas de Firebase para Android.

**+** *(Alternativa)* Agrega dependencias de la biblioteca de Firebase *sin* usar la BoM

Si eliges no usar la BoM de Firebase, debes especificar cada versión de la biblioteca de Firebase en su línea de dependencia.

**Ten en cuenta que, si usas *múltiples* bibliotecas de Firebase en tu app, es muy recomendable que utilices la BoM para administrar las versiones de las bibliotecas para garantizar que todas las versiones sean compatibles.**

```
dependencies {
    // Add the dependency for the Realtime Database library
    // When NOT using the BoM, you must specify versions in Firebase lib
    implementation("com.google.firebase:firebase-database:20.2.2")
}
```

## Configura reglas de seguridad de Realtime Database

Realtime Database proporciona un lenguaje de reglas declarativas que te permite definir cómo se deben estructurar los datos, cómo se deben indexar y cuándo se pueden leer y escribir.

**Nota:** Según la configuración predeterminada, el acceso de lectura y escritura a tu base de datos está restringido, por lo que solo los usuarios autenticados pueden leer o escribir datos. Para comenzar sin configurar Authentication (<https://firebase.google.com/docs/auth?hl=es-419>), puedes definir tus reglas para el acceso público ([https://firebase.google.com/docs/rules/basics?hl=es-419#default\\_rules\\_locked\\_mode](https://firebase.google.com/docs/rules/basics?hl=es-419#default_rules_locked_mode)). Esto hace que la base de datos esté abierta para todo el mundo, incluidas personas que no usan la app, así que asegúrate de volver a restringirla cuando configures la autenticación.

## Realiza operaciones de escritura en tu base de datos

Recupera una instancia de tu base de datos con `getInstance()` y haz referencia a la ubicación a la que quieras escribir.

**Importante:** Para obtener una referencia a una base de datos distinta de la predeterminada `us-central1`, debes pasar la URL de base de datos a `getInstance()` (o `database()` en Kotlin+KTX). Para obtener una base de datos predeterminada `us-central1`, puedes llamar a `getInstance()` (o `database`) sin argumentos.

Puedes encontrar la URL de Realtime Database en la sección *Realtime Database* de [Firebase console](https://console.firebase.google.com/?hl=es-419) ([//console.firebase.google.com/?hl=es-419](https://console.firebase.google.com/?hl=es-419)). Según la [ubicación de la base de datos](https://firebase.google.com/docs/projects/locations?hl=es-419#rtdb-locations) (<https://firebase.google.com/docs/projects/locations?hl=es-419#rtdb-locations>), la URL de la base de datos tendrá uno de los siguientes formatos:

- `https://DATABASE_NAME.firebaseio.com` (para bases de datos en `us-central1`)
- `https://DATABASE_NAME.REGION.firebaseio.com` (para bases de datos en todas las demás ubicaciones)

Kotlin+KTX

Java

Android

Android

```
// Write a message to the database
val database = Firebase.database
val myRef = database.getReference("message")

myRef.setValue("Hello, World!")
\src/main/java/com/google/firebase/referencecode/database/kotlin/MainActivity.kt#L28-L32)
```

De esta manera, puedes guardar diversos tipos de datos en la base de datos, incluso objetos de Java. Cuando guardas un objeto, las respuestas de cualquier método `get` se guardan como elementos secundarios de esta ubicación.

## Realiza operaciones de lectura desde la base de datos

Para que los datos de tu app se actualicen en tiempo real, agrega un [ValueEventListener](https://firebase.google.com/docs/reference/android/com/google/firebase/database/ValueEventListener?hl=es-419) (<https://firebase.google.com/docs/reference/android/com/google/firebase/database/ValueEventListener?hl=es-419>)

a la referencia que acabas de crear.

El método `onDataChange()` de esta clase se activa cuando se adjunta el objeto de escucha y cada vez que cambian los datos, incluidos los secundarios.

Kotlin+KTX

Java

Android

Android

```
// Read from the database
myRef.addValueEventListener(object : ValueEventListener {
    override fun onDataChange(dataSnapshot: DataSnapshot) {
        // This method is called once with the initial value and again
        // whenever data at this location is updated.
        val value = dataSnapshot.getValue<String>()
        Log.d(TAG, "Value is: $value")
    }

    override fun onCancelled(error: DatabaseError) {
        // Failed to read value
        Log.w(TAG, "Failed to read value.", error.toException())
    }
})
//src/main/java/com/google/firebase/referencecode/database/kotlin/MainActivity.kt#L36-L49
```

## Optional: Configura ProGuard

Cuando usas Firebase Realtime Database en tu app junto con ProGuard, debes considerar cómo se serializarán y deserializarán los objetos de tu modelo tras la ofuscación. Si usas `DataSnapshot.getValue(Class)` o `DatabaseReference.setValue(Object)` para leer y escribir datos, deberás agregar reglas al archivo `proguard-rules.pro` de la siguiente manera:

```
# Add this global rule
-keepattributes Signature

# This rule will properly ProGuard all the model classes in
# the package com.yourcompany.models.
# Modify this rule to fit the structure of your app.
-keepclassmembers class com.yourcompany.models.** {
    *;
}
```

Si quieres obtener ayuda con preguntas o problemas relacionados con ProGuard, visita los [foros de la comunidad de Guardsquare](https://community.guardsquare.com/?utm_source=site&utm_medium=site-link&utm_campaign=firebase-install-community)

([https://community.guardsquare.com/?utm\\_source=site&utm\\_medium=site-link&utm\\_campaign=firebase-install-community](https://community.guardsquare.com/?utm_source=site&utm_medium=site-link&utm_campaign=firebase-install-community))

para recibir asistencia de un experto.

## Prepárate para el lanzamiento

Antes de lanzar tu app, te recomendamos revisar nuestra [lista de tareas para el lanzamiento](https://firebase.google.com/support/guides/launch-checklist?hl=es-419) (<https://firebase.google.com/support/guides/launch-checklist?hl=es-419>) a fin de asegurarte de que tu app esté lista.

Asegúrate de habilitar la [Verificación de aplicaciones](https://firebase.google.com/docs/app-check/android?hl=es-419)

(<https://firebase.google.com/docs/app-check/android?hl=es-419>) para garantizar que solo tus apps puedan acceder a tus bases de datos.

## Próximos pasos

- Obtén información sobre cómo [estructurar datos](https://firebase.google.com/docs/database/android/structure-data?hl=es-419) (<https://firebase.google.com/docs/database/android/structure-data?hl=es-419>) para Realtime Database.
- [Escala tus datos en varias instancias de bases de datos](https://firebase.google.com/docs/database/usage/sharding?hl=es-419) (<https://firebase.google.com/docs/database/usage/sharding?hl=es-419>).
- [Lee y escribe datos](https://firebase.google.com/docs/database/android/read-and-write?hl=es-419) (<https://firebase.google.com/docs/database/android/read-and-write?hl=es-419>).
- [Consulta tu base de datos en Firebase console](https://console.firebase.google.com/project/_/database/data?hl=es-419) ([https://console.firebase.google.com/project/\\_/database/data?hl=es-419](https://console.firebase.google.com/project/_/database/data?hl=es-419)).

Salvo que se indique lo contrario, el contenido de esta página está sujeto a la [licencia Atribución 4.0 de Creative Commons](https://creativecommons.org/licenses/by/4.0/) (<https://creativecommons.org/licenses/by/4.0/>), y los ejemplos de código están sujetos a la [licencia Apache 2.0](https://www.apache.org/licenses/LICENSE-2.0) (<https://www.apache.org/licenses/LICENSE-2.0>). Para obtener más información, consulta las [políticas del sitio de Google Developers](https://developers.google.com/site-policies?hl=es-419) (<https://developers.google.com/site-policies?hl=es-419>). Java es una marca registrada de Oracle o sus afiliados.

Última actualización: 2023-09-22 (UTC)